MODE-PP HTML: A GDMO/GRM to HTML Translator -Release 1.0-Reference Manual

Olivier Festor

INRIA

INSTITUT˜NATIONAL˜DE˜RECHERCHE EN˜INFORMATIQUE˜ET˜EN˜AUTOMATIQUE

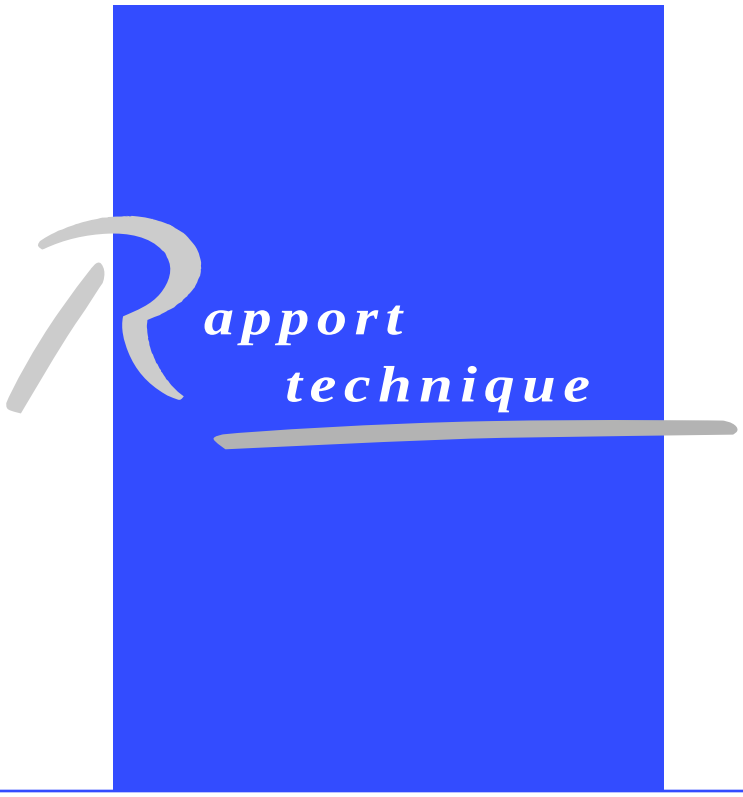
MODE-PP HTML: A GDMO/GRM to HTML translator -Release 1.0- Reference Manual

Olivier Festor

N ° 0199

décembre 1996

˜THÈME˜1˜



ISSN 0249-0803

MODE-PP HTML: A GDMO/GRM to HTML translator -Release 1.0- Reference Manual

Olivier Festor

Thème 1 — Réseaux et systèmes
Projet RESEDAS

Rapport technique n ° 0199 — décembre 1996 — ?? pages

Abstract: MODE-PP HTML (MODE Pretty-Printing HyperText Markup Language) is a module which generates a hypertext representation of GRM (General Relationship Model) and GDMO (Guidelines for the Definition of Managed Objects) specifications. The tool generates for a given specification the whole architecture required for the navigation facility within the specifications in an Intranet.

MODE-PP HTML is part of the MODERES (Managed Object Development Environment by RESEDAS) toolkit which is developed for facilitating the use of GDMO and GRM as well as integrating the use of Formal Description Techniques in an environment for building Management applications. MODE-PP HTML is built on top of MODE-FE and is provided as a library. This document is the reference manual for release 1.0 of the MODE-PP HTML library and its API.

Key-words: GDMO, GRM, MODE, HTML, Reference Manual

(Résumé : tsvp)

Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY (France)
Téléphone : (33) 03 83 59 30 30 – Télécopie : (33) 03 83 27 83 19
Antenne de Metz, technopôle de Metz 2000, 4 rue Marconi, 55070 METZ
Téléphone : (33) 03 87 20 35 00 – Télécopie : (33) 03 87 76 39 77

MODE-PP HTML: Un traducteur GDMO-GRM/HTML
-Version 1.0-
Manuel de référence

Résumé :

MODE-PP HTML (MODE Pretty-Printing HyperText Markup Language) est un module de génération de représentation hypertexte pour des spécifications GRM (General Relationship Model) et GDMO (Guidelines for the Definition of Managed Objects). Il génère pour une spécification donnée, toute l'archictecture nécessaire à la navigation hypertexte dans des modèles de l'information au sein d'un Intranet.

MODE-PP HTML est une application de l'environnement MODERES développé pour faciliter l'utilisation des spécifications GDMO et GRM ainsi que pour intégrer l'utilisation des méthodes de description formelle dans le développement d'applications de gestion de réseaux. MODE-PP HTML est construit sur la librairie MODE-FE. Il est disponible sous forme d'une librairie. Ce document forme le manuel de référence de la version 1.0 de la librairie MODE-PP HTML et de son interface de programmation.

Mots-clé : GDMO, GRM, MODE, HTML, Manuel de Référence

Chapter 1

Introduction

This document forms the reference manual of the **MODE-PP HTML** (release 1.0) part of the MODERES environment. MODERES stands for **Managed Object Development Environment by RESEDAS** and is a research prototype under development within the RESEDAS research group of the INRIA Lorraine and CRIN/CNRS. The MODERES environment is designed for supporting various developments based on latest OSI standards for Management Information Modeling combined with Formal Description Techniques.

The MODE-PP (MODE Pretty-Printer) is built on top of the MODE-FE Library [?]. MODE-PP provides formatting and pretty-printing facilities for both GRM (General Relationship Model) [?] and GDMO (Guidelines for the Definition of Managed Objects) [?] specifications. MODE-PP HTML is the HyperText Markup Language and Specification repository generator provided in the MODE-PP library.

Mode-PP is distributed with Mode-FE **FREELY** under the **COPYRIGHT** conditions as described in the `COPYRIGHT` file provided with the package and described in chapter ??. MODE is not a closed environment, nor it is a commercial one. Even if stable and complete, it aims at allowing people to experiment new features in Management Information Modelling.

Within this document, the features of the MODE-PP HTML generator are presented and the API detailed.

The remainder of this document is organized as follow. Chapter ?? provides an overview of the HTML generator's features. Chapter ?? presents the package and its use.

After some samples which illustrate some generated specifications, the document gives some information on contact and copyright concerning the MODE tool-set. The document is completed with a conclusion and a presentation of planned extensions (chapter ??). In an appendix, the reader will find a detailed description of the Application Programming Interface provided within the MODE-PP HTML.

Chapter 2

MODE-PP HTML overview

GDMO and GRM specifications are splitted into a number of sub-specifications which reference each other very often. Thus, having an overview of a complete specification is difficult and navigation facilities are of interest for that purpose. Moreover, providing a standardized repository which can be accessed within an Intranet by any person concerned with these specifications can be very helpful.

The provision of such an interface is the goal of the MODE-PP HTML part of the MODERES environment.

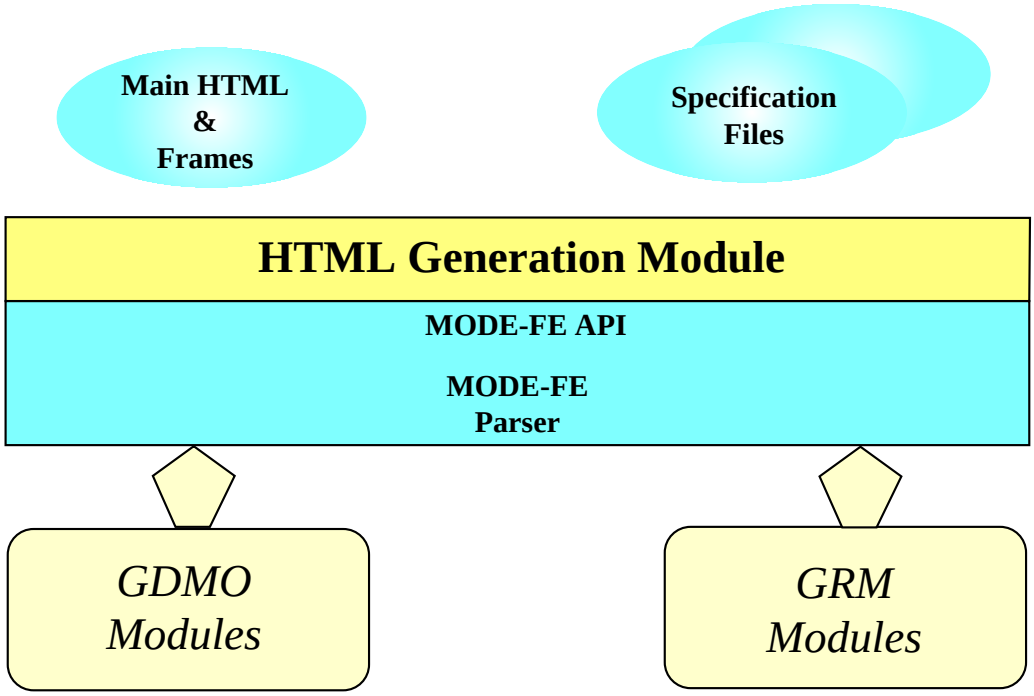


Figure 2.1: The MODE-PP Library and the HTML Generator: link to MODE-FE

The MODE-PP HTML generator is the second application module built over the MODE-FE toolkit. It uses the internal representation of specifications loaded with MODE-FE and provides HTML generation methods for one specification module.

Figure ?? illustrates the link between the HTML generator and the basic parts (i.e. the parser) of MODERES.

For one input specification, the generator creates a set of HTML files as well as all framing segments which form the repository for future navigation within the specifications using any browser that supports frames.

Chapter 3

Package Description, installation and usage

3.1 Installation

As MODE-PP-HTML is distributed within the MODERES package, see the installation commands in the MODERES documentation. GNUMAKE is required to compile the package.

To compile MODE-PP-HTML go to the `src` directory of the package, edit the `Makefile` and set the `CXX` variable to the name of the compiler you are using (CC, g++ or any C++ compiler).

Within the src directory, type `gmake -f Makefile`. This will build the object file of the library and put this file in the `lib` directory of the package.

3.2 Usage

The full HTML generator is defined in the MODE-PP library as one C++ class with many public printing methods. These methods are detailled in the appendix of the document. Each method is parameterized by the stream to which the output must be directed and a pointer to an object which contains the representation of the template(s) for which the method contains the HTML generation code.

Using the HTML printer requires:

- including all ModeFE related files,
- including the ModePP_HTMLPrinter.hh file in your program,
- having an instance of the parser, the repository and the message printer,
- having an instance of the HTML generator class,
- linking your applications with both the ModeFE and the ModePP libraries.

The code below illustrates the instanciation and a call to the generation of a HTML specification repository corresponding to a module previously parsed by the ModeFE parser.

MainHTML.cc

```
1 //
2 // MODE-FE
3 // Authors : Olivier Festor
4 // Emmanuel Nataf
5 //
6 // Copyright 1996 by Institut National de Recherche en Informatique
7 // et en Automatique (INRIA)
8 // All rights reserved. See COPYRIGHT in top-level directory.
9 //
```

```
10 // File: ModeFE_Mainparser.cc
11 // Description: VT100 Parser Implementation File.
12 // Revision: 1.0
13 // Contact: Olivier Festor
14 // INRIA Lorraine
15 // Technopole de Nancy-Brabois
16 // - Campus scientifique -
17 // 615, Rue du Jardin Botanique B.P. 101
18 // 54600 Villers-Les-Nancy Cedex
19 // FRANCE
20 // tel: (+33) 83.59.20.16
21 // fax: (+33) 83.27.83.19
22 // e-mail: festor@loria.fr
23
24 #include <stdio.h>
25 #include <iostream.h>
26 #include <fstream.h>
27 #include "ModeFE_AllClasses.hh"
28 #include "ModeFE_Grammar.bison.tab.h"
29 #include "ModePP_HTMLPrinter.hh"
30 #include "ModeFE_MessagePrinter.hh"
31
32 parse     Parser;
33 ModeFE_Repository  repository;
34 ModeFE_MessagePrinter messagePrinter;
35
36 int main(int argc, char **argv)
37 {
38  FILE *fp;
39  char* filename;
40  char* outfile;
41   // The HTML Generator
42  ModePP_HTMLPrinter pout;
43  int erreur;
44  if (argc != 2)
45    {
46      cout << endl << endl << "Usage: Main <grm-gdmo-file-name>" << endl << endl;
47      cout << "Try again!" << endl << endl;
48       // exit(1);
49    }
50  else
51    {
52      filename = strcopy(argv[1]);
53      if ((fp = freopen(filename,"r",stdin)) == NULL)
54        {
55          cout << " Could not open specification file " << endl;
56          return ERROR;
57        }
58      else
59        {
60          if ((erreur = Parser.yyparse()) == 0)
61            {
62              ofstream OutFile ( outfile, ios::out );
63              cout << "Parsing was Sucessful !!!" << endl;
64              cout << "...Nice job !!" << endl;
65              cout << "......You are an OSI Guru!!!!" << endl;
66              const ModeFE_ModuleList * ml = repository.CGetModules();
67              ModeFE_Module* res = ml->DGetElement();
68              pout.PrintHTMLModule(&OutFile,res);
69            }
70          else
71            {
```

```
72              cout << "Parsing was unsuccessful !!!" << endl;
73              cout << "...You should look at you GRM/GDMO specification!!" << endl;
74            }
75        }
76    }
77  cout << "Thanks for having used MODE-FE !\n\n\n\n";
78 }
79
```

Malte Zöckler

Chapter 4

Generated HTML files

For an input GRM/GDMO module, the back-end generates the following HTML files:

- `MODE_DEFAULT_BROWSER.html`: Navigation access file. This file must be loaded in the browser;
- `ModeRelCid.html`: label file for relationship classes. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeActionid.html`: label file for action classes. This file is used to set the list frames. It is automatically loaded by the browser;
- `Modepackid.html`: label file for packages. This file is used to set the list frames. It is automatically loaded by the browser;
- `Modemocid.html`: label file for managed object classes. This file is used to set the list frames. It is automatically loaded by the browser;
- `Modermapid.html`: label file for relationship mappings. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeParamid.html`: label file for parameters. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeAtgid.html`: label file for attribute groups. This file is used to set the list frames. It is automatically loaded by the browser;
- `Modenbid.html`: label file for name-bindings. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeBehid.html` : label file for behaviours. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeAttid.html` : label file for attributes. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeNotifid.html` : label file for notifications. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeMid.html`: label file for modules. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModePackages.html` : list of package specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeParameters.html` : list of parameter specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeMid.html`: list of managed object class specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeModules.html` : list of module specification labels. This file is used to set the list frames. It is automatically loaded by the browser. In this version of the back-end, only one module is supported;
- `ModeNbindings.html` : list of name-binding specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeNotifications.html` : list of notification specification labels. This file is used to set the list frames. It is automatically loaded by the browser;

- `ModeBehaviours.html` : list of behaviour specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `Modebehaviours.html`: list of behaviour specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `Modermappings.html`: list of relationship mapping specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeActions.html` : list of action specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeAttGroups.html` : list of attribute group specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeAttributes.html` : list of attribute specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeRelClasses.html`: list of relationship class specification labels. This file is used to set the list frames. It is automatically loaded by the browser;
- label file for relationship classes. This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeSpecSpace.html` : Default specification. Loaded during start when no specification appears in the right frame;
- `ModeRightLists.html`: This file is used to set the right part of the list frames. It is automatically loaded by the browser;
- `ModeLeftLists.html` : This file is used to set the left part of the list frames. It is automatically loaded by the browser;
- `ModeLists.html` : This file is used to set the list frames. It is automatically loaded by the browser;
- `ModeSpecs.html`: This file is used to divide the browser into a specification and a logo part;
- `ModeLogo.html`: This file is used to build the logo part of the browser;
- `xxx.html`: each GRM/GDMO template is generated into one html file.

All these files are generated in the directory in which the program was called.

Malte Zöckler

Chapter 5

Generation samples

Figure ?? contains a example of the browsing of GDM/GDMO generated navigation facilities. In this example, the Netscape browser is used. However, navigation within generated specification ma be done with any HTML browser that supports frames.

To load a specification repository, the user must load into his browser, the `MODE_DEFAULT_BROWSER.html` file previously generated. The load of this file results in the setting up of the screen as illustrated in figure ??. There the user may navigate within the specifications following the hypertext links.

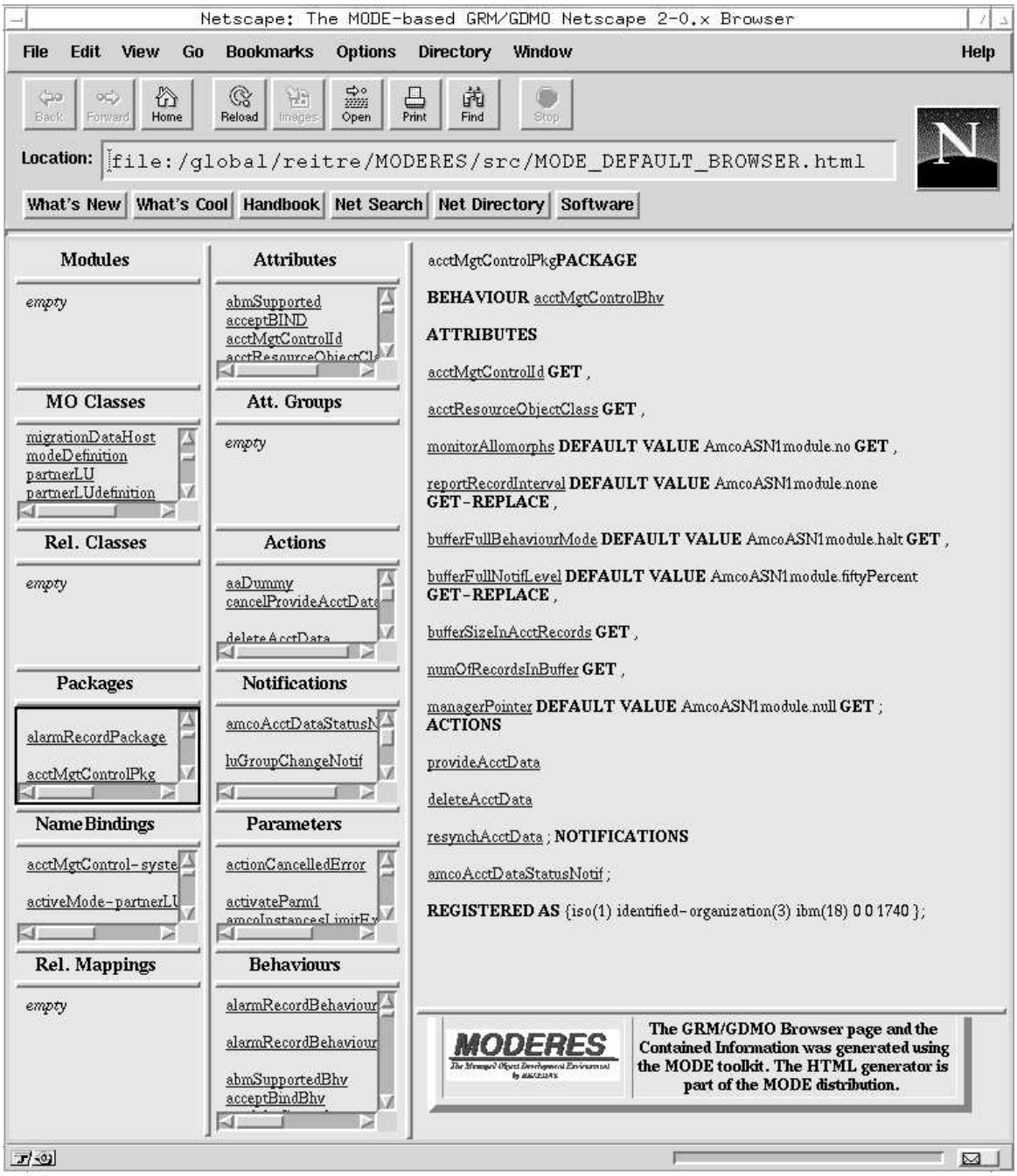


Figure 5.1: Browsing HTML generated specifications

Chapter 6

Copyright & Contact

6.1 Contact

The MODE toolkit is still under development. Several back-ends for MODE are either in development phase or β-test and some improvements to access the repository will be available in a very short time.

We are maintaining the MODE toolkit. So feel free to send us comments, bug reports. We will do our best to provide a stable and usage friendly tool.

Please send:

- bug reports,
- comments,
- any suggestions

to one of the authors. We will respond in the shortest delay.

Contact person
Olivier Festor
RESEDAS Research Group
INRIA Lorraine
Technopole de Nancy-Brabois
- Campus scientifique -
615, rue de Jardin Botanique - B.P. 101
54600 Villers Les Nancy Cedex
France
E-mail: festor@loria.fr
Tel: (33) 83.59.20.16
Fax: (33) 83.27.83.19
URL:http://www.loria.fr/~festor

A WWW page is maintained for MODE. The URL is:

http://www.loria.fr/exterieur/equipe/resedas/MODE.html

On this page you will find all new libraries developed for the the tool, new accessible applications, documentation, technical reports and white papers concerning planned extensions. You will also find many links to other network management pages from general information to companies which provide products for OSI and Internet Management.

6.2 Copyright

The ModePP library is distributed freely under the conditions specified in the COPYRIGHT file provided with the library and the MODE-FE toolkit. The copyright is as follows:

The ModePP library is distributed freely under the conditions specified in the COPYRIGHT file provided with the library. The reader will find below the content of the Mode-PP copyright.

The Software ModeFE (c) INRIA 1995 in its release 1.0 of the 16/04/1996, hereafter referred to as "The SOFTWARE".

The SOFTWARE has been designed and produced by O. Festor and E. Nataf, the researchers of the RESEDAS project, a research project of the National Computer And Automatics Institute (INRIA), Domaine de Voluceau, Rocquencourt, 78153 Le Chesnay Cedex.

INRIA holds all the patent rights concerning the SOFTWARE. The SOFTWARE has been registered at the Agency for the Protection of Programmes (APP) under the number:

IDDN.FR.001.190018.00.R.P.1996.000.10800

.

Foreword

The SOFTWARE is currently being developed and INRIA wishes for it to be used by the scientific world so as to test, evaluate and continually update it.

To this end, INRIA has decided to distribute the prototype of the SOFTWARE by FTP, in an Object-code form.

a) Extend of the rights granted by INRIA to the user of the SOFTWARE:

INRIA freely grants the right to use, modify and integrate the SOFTWARE in another program.

b) Reproduction of the SOFWARE

Clauses 9 and 10 of the Berne agreement for the protection of literary and artistic works (Union of Berne) respectively specify in their paragraphs 2 and 3 authorising only the reproduction and quoting of works on the condition that:

- *" This reproduction does not adversely affect the normal exploitation of the work or cause any unjustified prejudice to the legitimate interests of the author",*
- *"That the quotations given by way of illustration and/or tuition conform to the proper uses and that it mentions the source and name of the author if this name features in the source",*
- Any use or reproduction of the software items and/or documents exclusively owned by INRIA and carried out to obtain profit or for commercial ends being subject to obtaining the prior express authorisation of INRIA.
- Any commercial use made without obtaining the prior express agreement of the INRIA would therefore constitute a fraudulent imitation.

c) Information feed-back

Any user of the SOFTWARE shall send his comments on the use of the SOFTWARE to the INRIA at (email : festor@loria.fr).

d) Guarantees:

Note that the SOFTWARE is a research product currently being developed.

INRIA disclaims any responsibility in any way in any instance of being obliged to put right any possible direct or indirect damage sustained by the user.

Malte Zöckler

Chapter 7

Conclusion

In this report, we have presented the HTML generation module of the MODE toolkit. This module is built over the Mode-FE parser and allows the genration of browseable GRM and GDMO specifications. The module is, in its first release, still a prototype and may contain some minor errors.

Chapter 8

Planned extensions

Several extensions are planned for the MODE-PP library. Not all are definitively retained yet. Within these extensions we already identified:

- **Specification comments treatment and "As is" formatted printing:** We are currently extending the parser in a way that the comments written in a row GRM or GDMO specification are kept in the parser and a pretty printer for raw specifications (with comments and in-line specifications) is under development.
- **Graphical display of inheritance and containment relationships:** is planned in a near future. Stay tuned...

No major extentions except those described above are planned for the TeX printing module of ModeFE.

Bibliography

[CCITT.X.725 95] Comité Consultatif International Télégraphique et Téléphonique (CCITT), *Information Technology - Open Systems Interconnection - Structure of Management Information - Part 7: General Relationship Model*, International Standard, CCITT.X.725, November 1995.

[Festor 96] O.˜Festor, E.˜Nataf et L.˜Andrey. MODE-FE: A GRM/GDMO Parser and its API -Release 1.0- Reference Manual. Technical Report no.˜0190, INRIA Lorraine, 1996.

[ISO-10165.4 92] International Organization for Standardization (ISO), *Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects*, International Standard, ISO-10165.4, January 1992.

Appendix A

The API

Following methods are offered by any instance of a ModePP_TexPrinter class.

1

class **ModePP_HTMLPrinter**

Public Members

ModePP_HTMLPrinter ()
Basic constructor of the HTML Back-end. This methods generates the standard browser architecture and files

~**ModePP_HTMLPrinter** ()
Basic destructor of the HTML Back-end.

void **PrintHTMLModule** (ofstream*, const ModeFE_Module *)
This method prints the html code for one module and all associated files. The first parameter is the iostream to which errors are directed. The second parameter is a specification module

void **PrintManagedObjectClass** (ofstream*, const ModeFE_ManagedObjectClass*)
Generates the HTML code for one managed object class. The generated code is written onto the stream provided in the first parameter.

void **PrintPackage** (ofstream*, const ModeFE_Package*)
Generates the HTML code for one package. The generated code is written onto the stream provided in the first parameter.

void **PrintAction** (ofstream*, const ModeFE_Action*)
Generates the HTML code for one action. The generated code is written onto the stream provided in the first parameter.

void **PrintNotification** (ofstream* , const ModeFE_Notification*)
Generates the HTML code for one notification. The generated code is written onto the stream provided in the first parameter.

void **PrintParameter** (ofstream* , const ModeFE_Parameter*)
Generates the HTML code for one GDMO parameter. The generated code is written onto the stream provided in the first parameter.

void **PrintNameBinding** (ofstream*, const ModeFE_NameBinding*)
Generates the HTML code for one GDMO NameBinding. The generated code is written onto the stream provided in the first parameter.

void **PrintAttributeGroup** (ofstream* , const ModeFE_AttributeGroup*)
Generates the HTML code for one GDMO AttributeGroup. The generated code is written onto the stream provided in the first parameter.

void **PrintBehaviour** (ofstream*, const ModeFE_Behaviour*)
Generates the HTML code for one GDMO/GRM Behavior specification. The generated code is written onto the stream provided in the first parameter.

void **PrintAttribute** (ofstream*, const ModeFE_Attribute*)

Malte Zöckler

Generates the HTML code for one GDMO attribute specification. The generated code is written onto the stream provided in the first parameter.

void **PrintRelationshipClass** (ofstream*, const ModeFE_RelationshipClass*)
Generates the HTML code for one GRM Relationship Class. The generated code is written onto the stream provided in the first parameter.

void **PrintRelationshipMapping** (ofstream*, const ModeFE_RelationshipMapping*)
Generates the HTML code for one GRM Relationship Mapping. The generated code is written onto the stream provided in the first parameter.

Private Members

void **PrintLabel** (ofstream*, const ModeFE_Label*, int)
Private method. Generates the HTML code for one label to the provided stream. The integer parameter has the following meaning:

- *0: simple text*
- *1: generate associated hypertext link*

void **PrintLabelList** (ofstream*, const ModeFE_NameList*, char*, char *)
Private method. Generates the HTML code for a list of labels to the provided stream. The parameters has the following meaning:

- *1st char: separator which has to be printed*
- *2nd char: termination text*

void **PrintProperty** (ofstream*, const ModeFE_OneProperty*)
Private method. Prints one property associated to an attribute in a package definition.

void **PrintHeader** (ofstream*) *Private method. Generates the HTML header for a file to the stream provided in the parameter.*

void **PrintRelationshipOperation** (ofstream* Out , const ModeFE_RelationshipOperation* a)
Private method. Generates the HTML code related to a relationship Operation

void **PrintOperationMapping** (ofstream* Out, const ModeFE_SmoRoleList* a)
Private method. Generates the HTML Code related to an operation mapping. The first parameter is the stream to which the code is issued. The second is the operation mapping.

Malte Zöckler

Unit´e de recherche INRIA Lorraine, Technop^ole de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP˜101, 54600˜VILLERS˜LÈS˜NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042˜RENNES˜Cedex
Unit´e de recherche INRIA Rh^one-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP˜105, 78153˜LE˜CHESNAY˜Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP˜93, 06902˜SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP˜105, 78153˜LE˜CHESNAY˜Cedex (France)
http://www.inria.fr
ISSN 0249-6399